## Claims

1. (Currently Amended) A computer program product encoding a computer program for executing on a computer system a computer process for dynamically generating typing context data associated with a typing-context-relevant-code-point being executed within a typing context in a dynamic execution environment, the computer process comprising:

encountering the typing-context-relevant-code-point in the typing context during execution of the program;

identifying a typing context handle associated with the typing context, the typing context handle referencing a typing context data structure associated with the typing context;

computing the typing context data associated with the typing-context-relevant-code-point;

dynamically allocating a field in the typing context data structure associated with the typing-context-relevant-code-point; and

recording the typing context data in the field of the typing context data structure.

2. (Original) The computer program product of claim 1 wherein the typing-context-relevant-code-point executes a type test on an instance of a generic class, the typing context data includes a resource type descriptor defining the exact type of the instance, and the computer process further comprises:

performing the type test based on the resource type descriptor associated with the typing-context-relevant-code-point.

3. (Original) The computer program product of claim 1 wherein the typing-context-relevant-code-point executes an allocation of an instance of a generic class, the typing context data includes a resource type descriptor defining the exact type of the instance, and the computer process further comprises:

creating the instance of the generic class based on the resource type descriptor associated with the typing-context-relevant-code-point, wherein the instance is of the exact type.

4. (Original)  The computer program product of claim 1 wherein the typing-context-relevant-code-point calls a generic method, the typing context data includes another typing context handle, and the computer process further comprises:

passing the other typing context handle referencing the typing context data to the generic method as a hidden parameter.


5. (Original)  The computer program product of claim 1 wherein the identifying operation comprises:

retrieving the typing context handle from a stack frame.


6. (Original)  The computer program product of claim 1 wherein the typing-context-relevant-code-point is executed within an instance of a generic class and the identifying operation comprises:

retrieving a first pointer to the instance; and

retrieving the typing context handle via a second pointer, a second pointer being relative to the first point and referencing the typing context handle associated with the instance.

7. (Original) The computer program product of claim 1 wherein the computing operation comprises:

retrieving the typing context data associated with the typing-context-relevant-code-point from a global hash table.

8. (Original) The computer program product of claim 1 wherein the encountering operation comprises:

assigning an index to the typing-context-relevant-code-point.

9. (Original) The computer program product of claim 8 wherein the allocating operation comprises:

allocating the field in the typing context data structure, in accordance with the index.

10. (Original) The computer program product of claim 8 wherein the index is assigned based on the "arity" of the typing-context-relevant-code-point.

11. (Original) The computer program product of claim 8 wherein the

index is assigned based on a category associated with the typing-context-

relevant-code-point.


12. (Original) The computer program product of claim 11 wherein the

category is assigned on a per-containing class basis.


13. (Original) The computer program product of claim 11 wherein the

category is assigned on a per-containing method basis.


14. (Original) The computer program product of claim 11 wherein the

category is assigned on a per-containing assembly basis.


15. - 24.  Canceled


25. (Currently Amended) An execution engine for executing

parametrically polymorphic code and dynamically generating typing context

data associated with a typing-context-relevant-code-point being executed

within a typing context in a dynamic execution environment, the execution

engine comprising:

a read module configured to encounter ~~encountering~~ the typing-context-

relevant-code-point in the typing context during execution of the program;

a handle module configured to identify ~~identifying~~ a typing context

handle associated with the typing context, the typing context handle referencing

a typing context data structure associated with the typing context;

a computation module configured to compute ~~computing~~ the typing

context data associated with the typing-context-relevant-code-point;

an allocation module configured to dynamically allocate ~~allocating~~ a field

in the typing context data structure associated with the typing-context-

relevant-code-point; and

a recording module configured to record ~~recording~~ the typing context

data in the field of the typing context data structure.

26. (Currently Amended) A method of dynamically generating typing

context data associated with a typing-context-relevant-code-point being

executed within a typing context in a dynamic execution environment, the

method comprising:

encountering the typing-context-relevant-code-point in the typing

context during execution of the program;

identifying a typing context handle associated with the typing context,

the typing context handle referencing a typing context data structure associated

with the typing context;

computing the typing context data associated with the typing-context-

relevant-code-point;

dynamically allocating a field in the typing context data structure

associated with the typing-context-relevant-code-point; and

recording the typing context data in the field of the typing context data

structure.

27. – 36.  Canceled